



## 1.快速开发

用Python开发

数据库ORM系统



## 2.大量内置应用

后台管理系统 admin

用户认证系统 auth

会话系统 sessions



## 3.安全性高

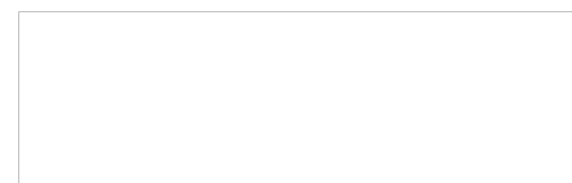
表单验证

SQL注入

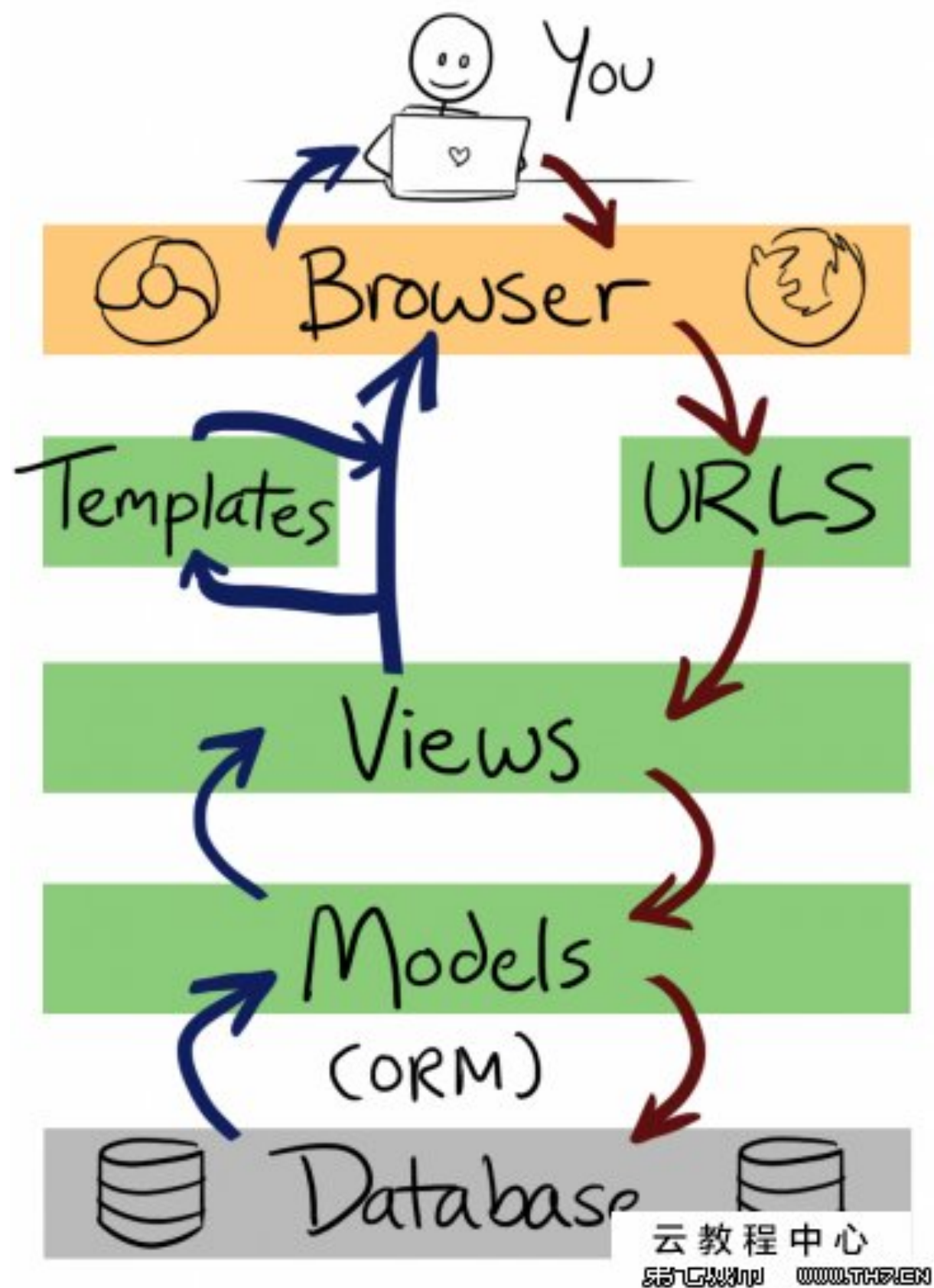
跨站点攻击

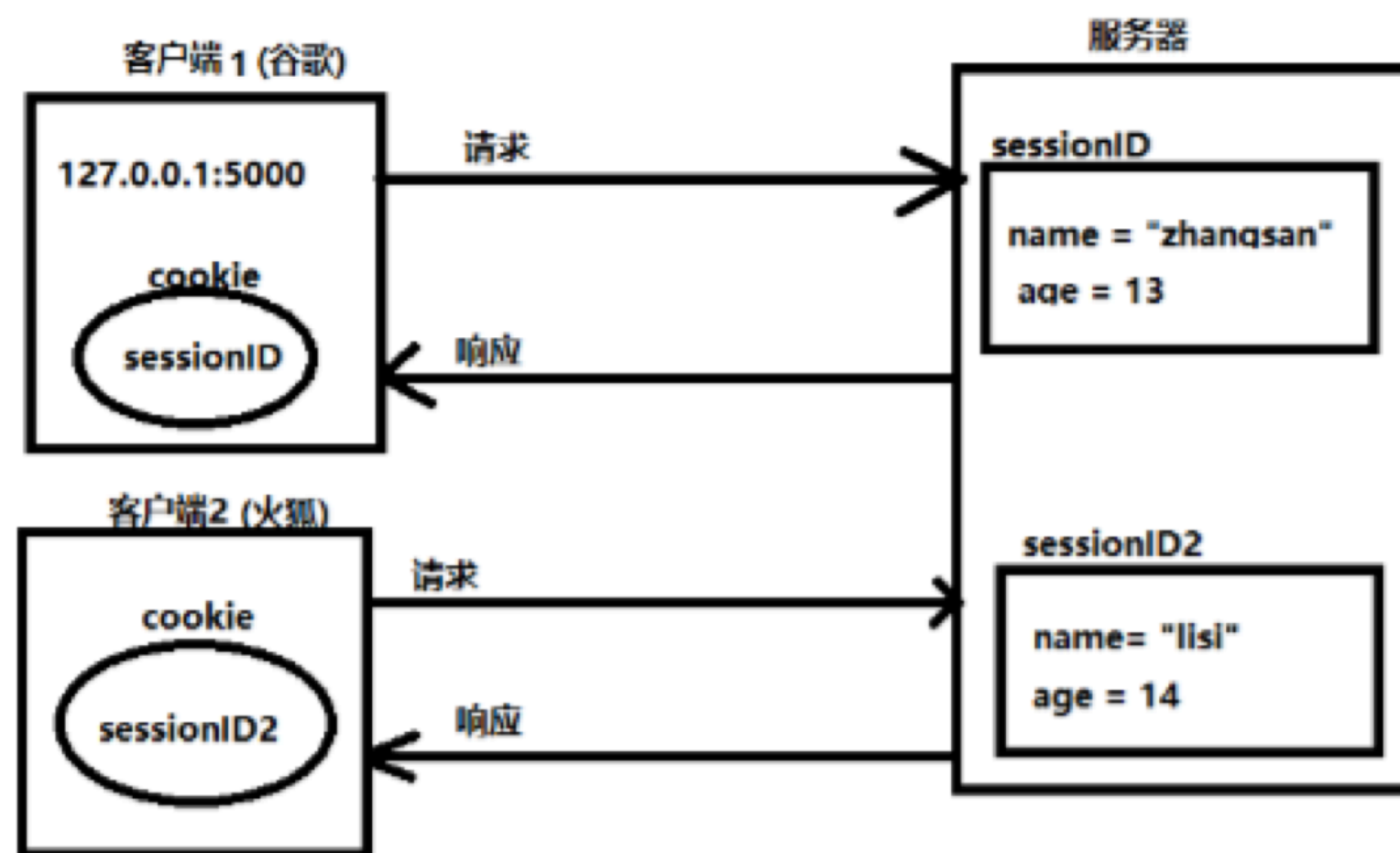


## 4.易于拓展



## 视图访问过程





session是用来存储客户端的一些隐私数据,保存在服务器中

session数据的设置过程:

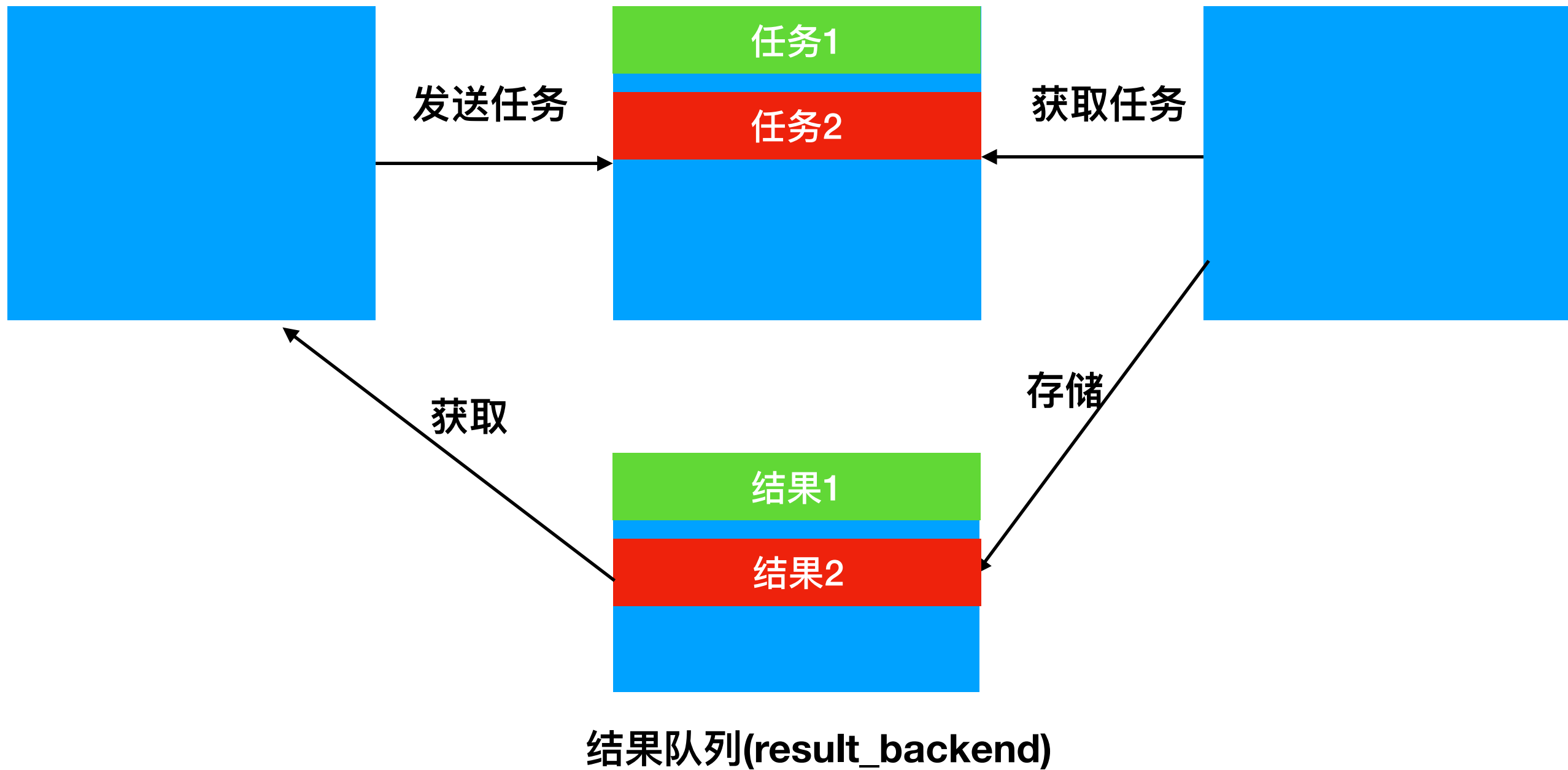
1. 当第一次请求的时候,在服务器会给客户端设置响应的session数据,返回给客户端,存储在cookie中
2. 当第二次发送请求的时候,cookie中会携带sessionID到服务端,通过id就可以获取到session中的数据了.

每一个用户向服务器请求的时候,都会为其开辟对应的空间保存session数据

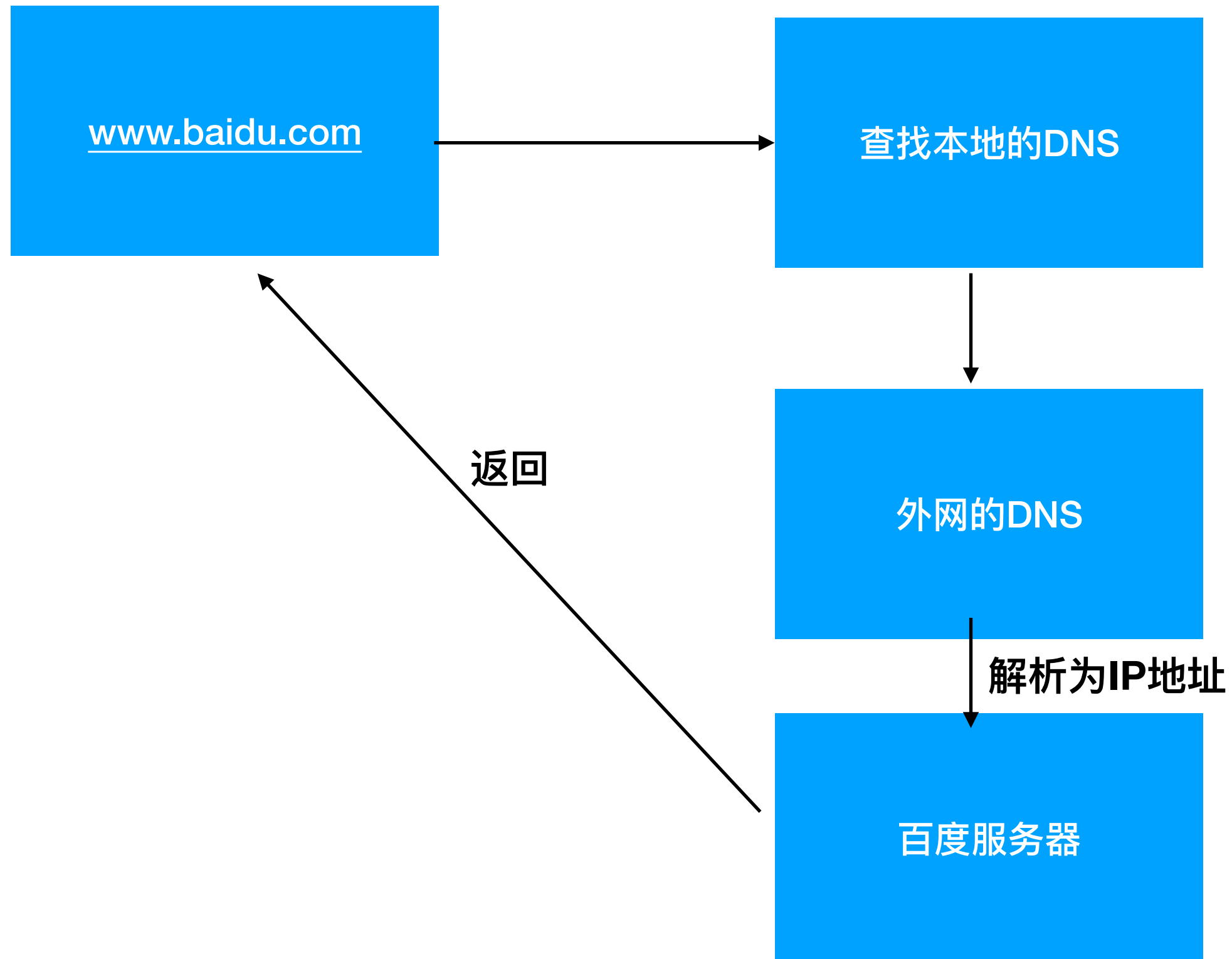
客户端

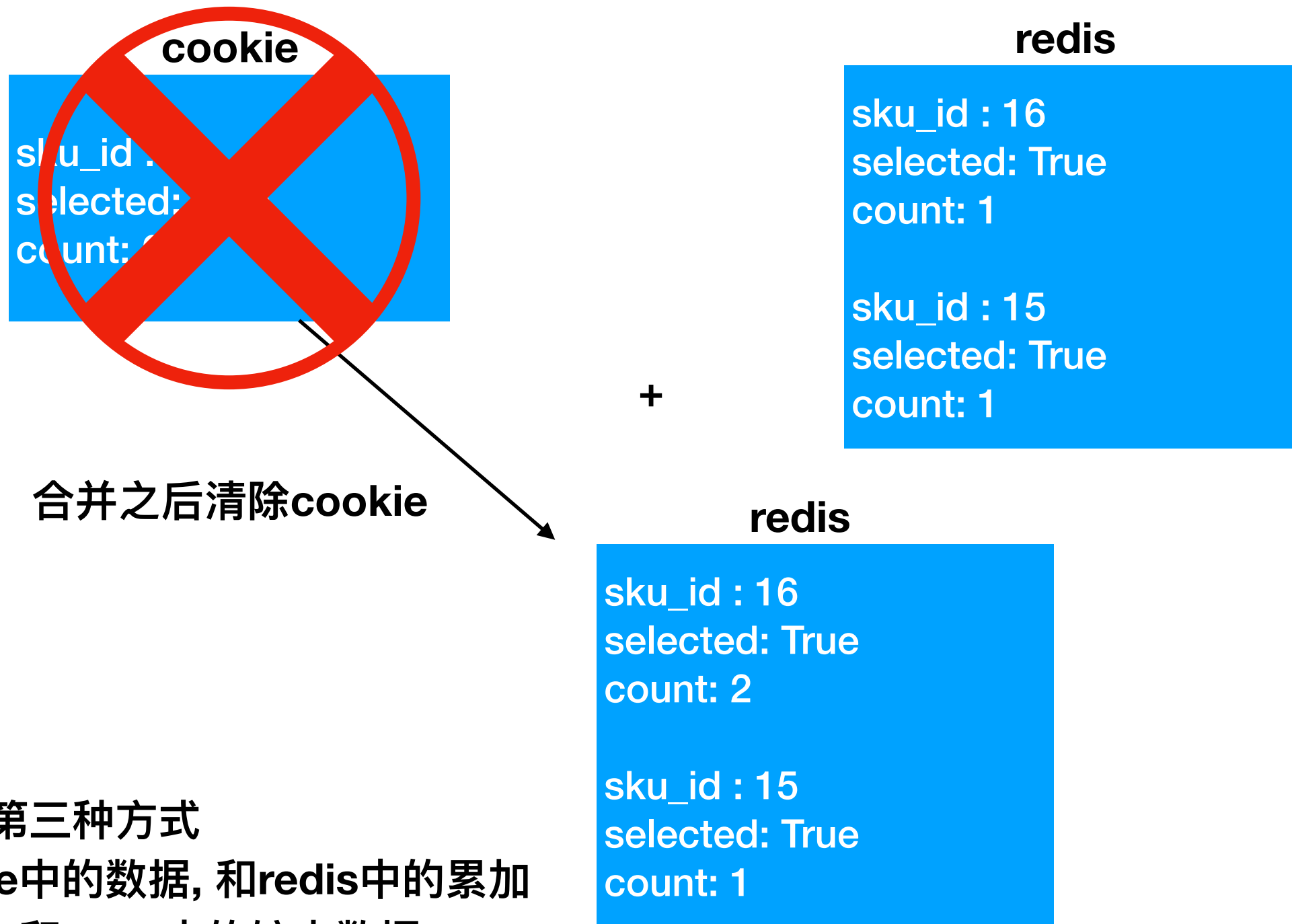
任务队列(broker)

任务处理者(worker)



# DNS





合并方式: 第三种方式

1, 将cookie中的数据, 和redis中的累加

2, 取cookie和redis中的较大数据

3, 直接使用cookie的数据, 覆盖redis的数据

部署方案: nginx + uwsgi + django

nginx: 静态服务器, 负载均衡, 分流, 转发等等

uwsgi: 业务服务器, 运行django代码

django: 提供代码的

